

### Claims

1. In a computer readable medium having plural object application processes that each include a separate execution model and a process memory for running programs with regard to encapsulated software objects, the improvement comprising:

a shared object memory in the computer readable medium storing encapsulated software objects that are directly accessible by the plural object application processes, the shared object memory not including an execution model and being distinct from the process memories of the object application processes; and

a shared object memory manager that provides management of objects within the shared object memory.

2. The medium of claim 1 in which the shared object memory manager provides garbage collection to remove unused objects in the shared object memory.

3. The medium of claim 1 further including an object namespace in the shared object memory listing software objects stored in the shared object memory.

4. The medium of claim 3 in which the shared object memory manager provides garbage collection to remove unused objects in the shared object memory according to whether the objects are referenced in the object namespace.

5. The medium of claim 3 in which the object namespace includes a data structure with a field ObjectName and a field ObjectID, in which the ObjectName field lists a name by which each object is accessed by an application process and the ObjectID field provides a reference for each object in an object table that includes a memory location pointer indicating a location where the object is located in the shared object memory.

6. The medium of claim 1 in which the shared object memory manager provides automatic creation in the shared object memory of each object that is referenced by a stored object.

7. The medium of claim 1 in which the shared object memory manager compacts the shared object memory when remaining free space becomes low.

8. The medium of claim 1 in which the shared object memory manager compacts the shared object memory when the amount of space reclaimed from objects that were garbage collected becomes high.

9. The medium of claim 1 in which each object in the shared object memory is of a class and the shared object memory manager provides registration of each class prior to an object of the class being stored in the shared object memory.

10. The medium of claim 1 in which the shared object memory manager provides for creation of objects in the shared object memory and initialization of object states with at least one of primitive data or reference to another object in the shared object memory.

11. The medium of claim 1 in which the object application processes include at least one Java virtual machine.

12. The medium of claim 1 in which the object application processes include at least one program in one of a family of C programming languages.

13. In a computer readable medium, shared object memory software for operating a shared object memory that is accessible by plural object application processes of a host computer that each include a separate execution model and a process memory for running programs with regard to encapsulated software objects, comprising:

software for allocating in the host computer a shared object memory that is distinct from the process memories of the object application processes;

software for creating software objects in the shared object memory;

software for providing the object application processes with direct access to the objects stored in the shared object memory; and

software for garbage collecting to remove unused objects in the shared object memory.

14. The medium of claim 13 in which the software for creating software objects in the shared object memory lists objects in an object namespace included in the shared object memory.

15. The medium of claim 14 in which the software for garbage collecting to remove unused objects in the shared object memory according to whether the objects are referenced in the object namespace.

16. The medium of claim 14 in which the object namespace includes a data structure with a field `ObjectName` and a field `ObjectID`, in which the `ObjectName` field lists a name by which each object is accessed by an application process and the `ObjectID` field provides a reference for each object in an object table that includes a memory location pointer indicating a location where the object is located in the shared object memory.

17. The medium of claim 13 in which the software for storing software objects in the shared object memory provides automatic storing in the shared object memory of each object that is referenced by a stored object.

18. The medium of claim 13 further comprising software for compacting the shared object memory when remaining free space becomes low.

19. The medium of claim 13 further comprising software for compacting the shared object memory when the amount of space reclaimed from objects that were garbage collected becomes high.

20. The medium of claim 13 in which each object in the shared object memory is of a class and the software for creating software objects in the shared object memory provides registration of each class prior to an object of the class being stored in the shared object memory.

21. The medium of claim 13 in which the software for creating software objects in the shared object memory provides for creation of objects in the shared object memory and initialization of object states with at least one of primitive data or reference to another object in the shared object memory.

22. The medium of claim 13 in which the object application processes include at least one Java virtual machine.

23. The medium of claim 13 in which the object application processes include at least one program in one of a family of C programming languages.